

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-007407

(43)Date of publication of application : 12.01.1999

(51)Int.Cl. G06F 12/00  
G06F 13/00

(21)Application number : 10-117344 (71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 27.04.1998 (72)Inventor : BRENT ZION HAIRUPAAN  
PETER KENNETH MALKIN  
ROBERT JEFFREY SHUROSS  
YU PHILIP S

(30)Priority

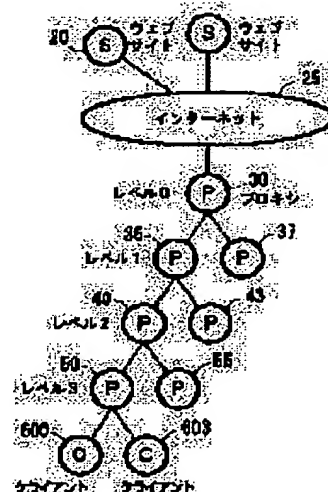
Priority number : 97 854226 Priority date : 09.05.1997 Priority country : US

(54) DYNAMIC PUSH FILTERING METHOD ACCOMPANIED BY STAGING/  
BUFFERING IN PROXY HIERARCHY

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a push base filtering system in a client-server hierarchy and a method for annexing the annotation of meta information related to the contents of a push object or urgency or both of them to the object based on using state information.

**SOLUTION:** Objects are staged in a server so that each filtered object can be quickly accessed at the time of requiring it thereafter. A contents provider or an upper level proxy for annotating an object transmits information including urgency, a summary or a title, group classification, or push discrimination, or all of them by using a PICS protocol so that using state or preference information related to the pushed object including using state information and user preference based on object group classification is transmitted to an upper level in hierarchy and each staged object is transmitted to a lower level in the hierarchy, improving the efficiency of caching.



## LEGAL STATUS

[Date of request for examination] 23.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2893019

[Date of registration] 26.02.1999

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-7407

(43) 公開日 平成11年(1999) 1月12日

(51) Int.Cl.<sup>9</sup>

識別記号

F I

G 0 6 F 12/00  
13/00

5 4 6  
3 5 5

G 0 6 F 12/00  
13/00

5 4 6 K  
3 5 5

審査請求 有 請求項の数27 O L (全 18 頁)

(21) 出願番号 特願平10-117344

(22) 出願日 平成10年(1998) 4月27日

(31) 優先権主張番号 0 8 / 8 5 4 2 2 6

(32) 優先日 1997年 5月 9日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー  
ズ・コーポレーション

INTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72) 発明者 プレント・ツイオン・ハイルバーン

アメリカ合衆国10536-2010 ニューヨー  
ク州カトナ ザ・テラス 26

(74) 代理人 弁理士 坂口 博 (外1名)

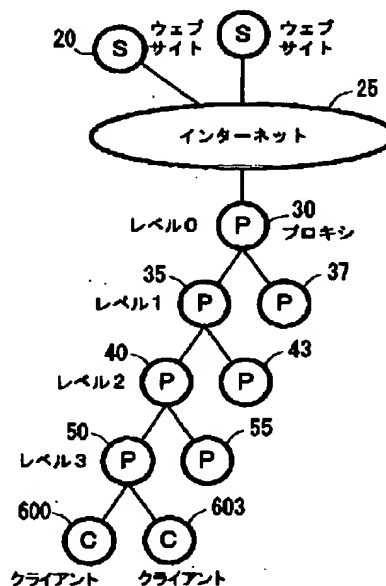
最終頁に続く

(54) 【発明の名称】 プロキシ階層におけるステージング/バッファリングを伴う動的プッシュ・フィルタリング方法

(57) 【要約】

【課題】 利用状況情報に基づきクライアントーサーバ階層でのプッシュ・ベースのフィルタリング方式と、プッシュ・オブジェクトにオブジェクトのコンテンツまたは緊急度あるいはその両方に関するメタ情報の注釈を付ける方法を提供すること。

【解決手段】 オブジェクトをサーバ上でステージングし、フィルタリングされたオブジェクトが後に要求された時点で高速アクセスを可能にすることができる。P I C Sプロトコルを使って、コンテンツ・プロバイダまたはオブジェクトに注釈を付ける上位レベル・プロキシによって、緊急度、サマリまたはタイトル、グループ分類、もしくはプッシュの識別あるいはこれらのすべてを含む情報を送信し、オブジェクト・グループ分類に基づく利用状況情報およびユーザ・プリファレンスを含む、プッシュされたオブジェクトに関する利用状況またはプリファレンス情報を階層中を上方に送信し、それぞれのステージングされたオブジェクトを階層中を下方に送信してキャッシングの効率を向上させることができる。



プロキシ・サーバの階層

## 【特許請求の範囲】

【請求項 1】プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてその後プッシュされたオブジェクトを動的にフィルタリングする方法であって、

プッシュされたオブジェクトに関連する利用状況情報を階層中を上方に送信するステップと、  
送信されたオブジェクト利用状況情報に基づいてその後プッシュされたオブジェクトをフィルタリングするステップとを含む方法。

【請求項 2】プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてプッシュされたオブジェクトを動的にフィルタリングする方法であって、情報を総合しノード間で交換するステップと、プッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングするステップとを含む方法。

【請求項 3】前記フィルタリング・ステップがフィルタリングされたオブジェクトに関連するメタ情報を送信するステップをさらに含む、請求項 2 に記載の方法。

【請求項 4】フィルタリング決定、選択された下位レベル・プロキシ／クライアントへのプッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップをさらに含む、請求項 1 に記載の方法。

【請求項 5】プッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後プッシュされたオブジェクトをフィルタリングする前記ステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む、請求項 1 に記載の方法。

【請求項 6】メタ情報を送信する前記ステップが、オブジェクトの短い説明を階層中を下方に送信するステップと、  
フィルタリングされたオブジェクトをプロキシ階層内でステージングするステップとをさらに含む、請求項 5 に記載の方法。

【請求項 7】オブジェクトをステージングする前記ステップが、  
ユーザ・プロファイルで示された潜在的なインタレストを有するすぐ下位のレベルのすべてのプロキシまたはクライアント・ノードがオブジェクトを受信したか、あるいはステージング緊急度因数が所定および計算しきい値を下回ったか、そのいずれかの後にステージングされたオブジェクトをバージするステップをさらに含む、請求項 4 に記載の方法。

【請求項 8】前記フィルタリング・ステップがさらに階層内のすべての下位クライアント・ノードによるプッシュされたオブジェクトの総合利用状況およびプリファレンスを規定するステップをさらに含む、請求項 1 に記載

の方法。

【請求項 9】オブジェクトがオブジェクト・グループに分類され、各オブジェクト・グループの利用状況情報が以前のユーザ要求パターンに基づく、請求項 4 に記載の方法。

【請求項 10】利用状況情報が下位レベル・プロキシまたはクライアントの要求パターンを含み、  
前記プッシュされたオブジェクトを異なるグループに分類するステップと、

プッシュされたオブジェクトのグループ分けを含む利用状況情報を総合しノード間で交換するステップと、  
総合され交換された情報およびプッシュされたオブジェクトのグループ化に基づいてプッシュされたオブジェクトをフィルタリングするステップとをさらに含む、請求項 1 に記載の方法。

【請求項 11】フィルタリング決定またはステージング決定が帯域幅、オブジェクト・プロパティ、またはクライアント特性のいずれかの関数である、請求項 4 に記載の方法。

【請求項 12】クライアント特性がユーザ・プロファイルまたはプリファレンス情報を含む、請求項 11 に記載の方法。

【請求項 13】オブジェクト緊急度標識をプッシュされたオブジェクトに関連付けて階層中を下方に送信するステップをさらに含む、  
前記フィルタリング・ステップがオブジェクト緊急度標識の関数である、請求項 1 に記載の方法。

【請求項 14】緊急度標識を 1 つまたは複数の異なる階層レベルに関連付けるステップをさらに含む、  
前記ステージング・ステップが、緊急度標識の関数としてコンテンツ階層の 1 つまたは複数のレベルでオブジェクトをステージングするステップを含む、請求項 4 に記載の方法。

【請求項 15】前記フィルタリング・ステップとステージング・ステップのいずれかがオブジェクト・サイズの関数である、請求項 4 に記載の方法。

【請求項 16】前記フィルタリング・ステップと前記ステージング・ステップのいずれかがオブジェクトの寿命または満了時間の関数である、請求項 4 に記載の方法。

【請求項 17】前記ステージング・ステップにตอบสนองして、プッシュされたオブジェクトのステージング・ステータスをプッシュされたオブジェクトへ送信するステップをさらに含む、請求項 4 に記載の方法。

【請求項 18】プロキシ階層が、前記フィルタリング・ステップと前記ステージング・ステップのいずれかが階層内のすべてのサーバによっては実行されない異種プロキシ階層を含む、請求項 4 に記載の方法。

【請求項 19】P I C S 利用状況ラベルを作成し、P I C S カテゴリ値を使ってプロキシ階層の下位レベルでの総合利用状況情報を表すステップをさらに含む、

前記送信ステップが P I C S 利用状況ラベルを使って総合利用状況情報を階層中を上方に送信するステップを含む、請求項 2 に記載の方法。

【請求項 20】 P I C S ステージング・ラベルを作成し、P I C S カテゴリ値を使ってプロキシ階層の所与のレベルにおけるステージングされたオブジェクトのステージング・ステータスを表すステップと、P I C S ステージング・ラベルを使ってステージング・ステータスを階層中を下方に送信するステップをさらに含む、請求項 4 に記載の方法。

【請求項 21】 P I C S プッシュ・ラベルを作成し、P I C S カテゴリ値を使ってプッシュされたオブジェクトの緊急度標識を表すステップと、P I C S プッシュ・ラベルを使って緊急度標識を階層中を下方に送信するステップとをさらに含む、請求項 4 に記載の方法。

【請求項 22】 メタ情報プロトコルを使って階層中を情報を送信する、請求項 1 ないし 3、5、6、9 ないし 10、14 ないし 15、19 ないし 20 または 21 のいずれか一項に記載の方法。

【請求項 23】 P I S C プロトコルを使って階層中を情報を送信する、請求項 1 ないし 3、5、6、9 ないし 10、14 ないし 15、19 ないし 20 または 21 のいずれか一項に記載の方法。

【請求項 24】 P I C S プッシュ・ラベルを作成し、P I C S カテゴリ値を使ってプッシュされたオブジェクトのサマリを表すステップと、P I C S プッシュ・ラベルを使ってサマリを階層中を下方に送信するステップとをさらに含む、請求項 4 に記載の方法。

【請求項 25】 階層外のプロキシ・ノードについてのステージング決定に基づいてオブジェクトの適応ステージングを行うステップをさらに含む、請求項 1 に記載の方法。

【請求項 26】 プッシュされたオブジェクトのストリームが階層中を下方に送信されるプロキシ階層において、その後プッシュされるオブジェクトを動的にフィルタリングする方法であって、プッシュ・ストリームにメタ情報の注釈を付けるステップと、前記の注釈付けステップに応答して、1 つまたは複数のプッシュされたオブジェクトをフィルタリングするステップとを含む方法。

【請求項 27】 フィルタリング決定とステージング決定のいずれかが、コンテンツ階層のレベルが異なると変わる、請求項 6 に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、改良されたデータ処理システムに関する。本発明の特定の態様は、プロキシ・サーバの階層内でオブジェクトまたはドキュメントを送達する動的プッシュ（または同報通信）フィルタリング方式に関する。本発明のさらに特定の態様は、ウェブ・オブジェクトをワールド・ワイド・ウェブ（WW

W）にプッシュする手順に関する。

【0002】用語集

本明細書で使用する用語の一部は辞書にある意味もあるが、次の用語集も役立つであろう。

【0003】インターネット

一連の TCP/IP プロトコルを使用するネットワークおよびゲートウェイからなるネットワーク。

【0004】クライアント

クライアントとは、コマンドをサーバに発行してそのコマンドに対応するタスクを実行させるコンピュータである。

【0005】サーバ

他のコンピュータのコマンドに従ってタスクを実行する任意のコンピュータがサーバである。ウェブ・サーバは通常 1 つまたは複数のクライアントをサポートする。

【0006】ワールド・ワイド・ウェブ（WWW またはウェブ）

強調表示された言葉や関心がある語句（ハイパーリンク）をクリックすることにより、サーバ間およびデータベース間でインターネット・スイッチ上で情報を探すためのインターネットのアプリケーション。インターネット WWW サーバはクライアントをサポートし、情報を提供する。ウェブはすべての資源が URL としてアドレス指定され、HTML を使って URL に対応する情報を表示し、他の URL へのポイント・アンド・クリックによるインタフェースを提供するインターネットと見なすことができる。

【0007】ユニバーサル・リソース・ロケータ（URL）

インターネット上の情報を一意的に識別またはアドレス指定するための方法。Eメール・アドレスのウェブ・ドキュメント・バージョンまたは完全に資格があるファイル名。これらはハイパーリンクでアクセス可能である。URL の 1 例は、「http://www.philipyu.com:80/table.html」である。ここで URL は 4 つの成分を備える。左から始めて、最初は使用するプロトコルを指定するもので、残りのロケータと「:」で分離される。2 番目は目標ホストのホスト名または IP アドレスである。これは左側が「/」で、右側が「/」または任意選択として「:」で区切られる。ポート番号は任意選択で、左側はホスト名と「:」で区切られ、右側は「/」で区切られる。第 4 の成分は実際のファイル名またはプログラム名である。この例では、「.html」の拡張子が HTML ファイルであることを示している。

【0008】ハイパーテキスト・マークアップ言語（HTML）

HTML はとりわけ、ドキュメントを作成し、ウェブ・クライアントから閲覧できる他のウェブ・ドキュメント

に（ハイパーリンク経由で）接続するためにウェブ・サーバが使用する言語である。

【0009】ハイパーテキスト転送プロトコル（HTTPまたはhttp）

HTTPはクライアントからサーバへのすべての要求が独立して扱われることを意味する無状態プロトコルの例である。サーバは以前の接続の記録を持たない。URLの冒頭にある「http:」は要求元クライアントと目標サーバが指定の資源に関してHTTPを使って通信を行うことを示している。

【0010】インターネット・ブラウザまたはウェブ・ブラウザ

httpなどのインターネット・プロトコルを実行し、その結果をユーザの画面に表示するグラフィカル・インタフェース。ブラウザは、ユーザがインターネットの「サーフィン」をする際に使用されるデスクトップ画面、ディレクトリおよび検索ツールを備えたインターネットのツアー・ガイドとして機能できる。この適用例ではウェブ・ブラウザはワールド・ワイド・ウェブと通信するクライアント・サービスである。

【0011】クライアント・キャッシュ

クライアント・キャッシュは通常、クライアントがアクセスするオブジェクト用の1次キャッシュとして使用される。WWW環境では、クライアント・キャッシュは通常ウェブ・ブラウザによって実施され、現在の呼出し中にオブジェクトをキャッシュする非持続性キャッシュでもよく、複数の呼出しにまたがってオブジェクトをキャッシュすることもできる。

【0012】キャッシング・プロキシ

クライアントのためにエージェントとして働いて、オブジェクトのキャッシュされたコピーを見つける、ネットワーク内の専用サーバ。キャッシング・プロキシは、クライアント・キャッシュからのキャッシュ・ミスの結果として呼び出されるため、通常は2次またはそれ以上のレベルのキャッシュとして動作する。

【0013】HTTPデーモン（HTTPD）

ハイパーテキスト転送プロトコルおよび共通ゲートウェイ・インタフェース機能を備えるサーバ。HTTPDは通常、イントラネット上のマシンへのハードウェア接続、およびTCP/IPカブリリングなどインターネットへのアクセスを提供する、アクセス・エージェントによってサポートされる。

【0014】ワールド・ワイド・ウェブ（WWWまたはウェブ）の人気の急激な高まりに伴って、インターネット上のトラフィック量も増加している。その結果、ウェブはネットワーク・パフォーマンスの主要なボトルネックになってきた。低速ネットワーク・リンクを介してサーバに接続されているユーザからドキュメントまたは情報の要求があった場合、ユーザ・エンドではかなりの待ち時間を覚悟しなくてはならない。要求されたドキュメ

ントを「プル」するのに長時間待つという手間を避けるための代替方法は、コンテンツ・プロバイダに該当するドキュメントが使用可能になると同時にあらかじめ規定したユーザ・プリファレンスまたはプロファイルに基づいてユーザヘッドキュメントを「プッシュ」させる方法である。

【0015】このプッシュ方法は、ネットワークからオーバフローする傾向がある。これは、ユーザのプリファレンス指定が不適切になりがちなため、あまりにも多くのドキュメントがユーザの元にプッシュされることが原因で起こる。

【0016】従来の「プル」方法では、アクセスの待ち時間を減らす1つの方法は人気があるドキュメントまたはユーザに身近な情報のコピーをキャッシュすることである。ここからのアクセスの待ち時間はより短くなる。このキャッシングはネットワーク上のさまざまなポイントで実施できる。例えば、大きな大学や会社では独自のローカル・キャッシュがあり、そこからネットワークの加入したすべてのユーザがドキュメントを取り出すことができる場合がある。場合によっては、クライアントのためのエージェントとして動作できる、キャッシング・プロキシと呼ばれる専用サーバが、ドキュメントのキャッシュされたコピーを見つけるためにネットワーク中で実施される。通常、キャッシング・プロキシは（1次）クライアント・キャッシュからのキャッシュ・ミスにのみ関連しているため、2次またはそれ以上のレベルのキャッシュとして動作する。クライアント・キャッシュは通常ウェブ・ブラウザの一部であり、現在の呼出し中にアクセスされたオブジェクトを記憶する（Mosaicで実施されるような非持続性キャッシュ）、または、複数の呼出しにまたがってアクセスされたドキュメントを記憶することができる。

【0017】一般的に、プロキシの階層はクライアントとサーバ（1つまたは複数の）によって構成される。コンピュータ・ネットワークでは、プロジェクト・プロキシ、部門プロキシ、サイト・プロキシなどが1つまたは複数存在する。インターネットのサービス・プロバイダは各近隣、各下位地域、各地域などの1つまたは複数でプロキシを実施できる。クライアントまたはプロキシあるいはその両方がキャッシング階層を形成する。厳密な階層では、キャッシュ・ミスが生じると（クライアント）またはプロキシは、CERN HTTPキャッシュで使われるHTTPインタフェースのようなキャッシング・プロキシ・インタフェースを介して階層のすぐ上位レベルから欠落したオブジェクトを要求する。より最近では、ハーベストにおいて、キャッシュ・ミス時に「兄弟」または「近隣」キャッシュに問い合わせることができる（C. M. ブラウン（Brown）他、「Harvest: A Scalable, Customizable Discovery and Access System」コロラド大学、コンピュータ科学部、技術レポート

CU-CS-732-94、1994年を参照)。いずれの場合も、キャッシングの決定は他のプロキシでキャッシングされたオブジェクトとは無関係にそれぞれのローカル・プロキシで行われる。言い換えると、キャッシングの決定はローカル・キャッシュのコンテンツまたはオブジェクト特性あるいはその両方の関数としてのみ実行される。

#### 【0018】

【発明が解決しようとする課題】以上のように、ブッシュ・ベースのフィルタリングおよびプロキシ・サーバを開発し、ビューワの実際の利用状況に基づくシステムが必要である。さらに、ステージング決定がブッシュ・フィルタリング決定およびブッシュ・アクティビティの結果に基づいて実行されるシステムおよび方法が必要である。また、プロキシ・サーバ間、コンテンツ・サーバとクライアント間での情報の送受信または交換によってプロキシ階層をより効果的に運用する必要がある。本発明は上記の必要に関する。

#### 【0019】

【課題を解決するための手段】上記の必要に応じて、本発明は、実際の利用情報に基づくクライアントサーバ階層でブッシュ情報をフィルタリングする方法およびシステムを対象とする。利用状況情報は実際のオブジェクト参照/アクセス・パターンを含むことができる。ステージング・キャッシュ・マネージャはサーバ（1つまたは複数）で実施され、フィルタリングされたオブジェクトが後で要求された時点で高速アクセスを可能にできる。

【0020】ブッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてブッシュされたオブジェクトを動的にフィルタリングする本発明による機能を備えた方法は、ブッシュされたオブジェクトに関連する利用情報を階層中を上方に送信するステップと、その後、送信されたオブジェクト利用情報に基づいてブッシュされたオブジェクトをフィルタリングするステップを含む。

【0021】本発明の別の態様は、情報を総合してノード間で交換する追加ステップと、ブッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングする追加ステップを含む。このフィルタリング・ステップは、さらにフィルタリングされたオブジェクトに関連するメタ情報を送信するステップを含むことができる。

【0022】本発明のさらに別の態様は、フィルタリング決定、選択された下位レベル・プロキシ/クライアントへのブッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップを含む。その他の機能によってプロキシ・サーバはステージングされたオブジェクトをタイムリにページし、オブジェクトの冗長ステージングを削減する

ことによってより有効にキャッシュを管理することができる。

【0023】本発明のさらに別の態様では、ブッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後ブッシュされたオブジェクトをフィルタリングするステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む。

【0024】本発明の別の態様では、ブッシュ・フィルタリングはコンテンツ・レベルでのブッシュの決定を含む。フィルタリングの決定はタイトルまたはサマリ（コンテンツ全体ではなく）だけを次の（下位の）レベルの階層のいくつかのノードへブッシュすることでもよい。このように、フィルタリングの決定は次のどの（下位の）レベルのノードがブッシュを受信するかだけでなく、各ノードが受信するコンテンツ・レベルで可能である。次の（下位の）レベルのノードへのフィルタリングの決定はその次の（下位の）レベルのノードの下すべてのユーザの利用状況の総合情報に基づくことができる。

【0025】本発明のさらに別の態様では、ブッシュされたオブジェクトのストリームが階層中を下方に送信され、その後ブッシュされたオブジェクトを動的にフィルタリングする方法が、ブッシュ・ストリームにメタ情報の注釈を付けるステップと、注釈付けのステップに回答してブッシュされた1つまたは複数のオブジェクトをフィルタリングするステップを含む。

【0026】本発明のさらに別の態様では、インターネット環境で、PICSプロトコルを使って各種の情報を送信できる。まず、コンテンツ・プロバイダまたは上位レベルのプロキシ・ノードはPICSを使ってオブジェクトに注釈を付ける、すなわち、ブッシュ・オブジェクトに関する情報の特性を指定できる。この情報はブッシュ・オブジェクトの緊急度または優先度、オブジェクト・コンテンツのサマリまたはタイトル、グループ分類、またはブッシュの送信元チャネル（識別）あるいはこれらすべてを含むことができる。第2に、PICSを使ってブッシュ・オブジェクトのプリファレンス情報の利用状況を階層の下位レベルから上方に送信できる。これにはオブジェクト・グループ分類に基づく利用状況情報およびユーザ・プリファレンスを含むことができる。第3に、PICSを使って、それぞれのステージングされたオブジェクト（上位階層のいくつかまたはすべての）のステージング状態を階層中を下方へ送信することができる。この場合、PICSプロトコルを一般化して階層全体にわたって情報の交換または送信を行うことができる。より詳細に言えば、これらの各種の情報はPICSプロトコルを使ってオブジェクトのヘッダに記憶することができる。各情報タイプについて新しいPICSラベルを定義してPICSカテゴリ値を特定の状況に対応させることができる。下位レベル・サーバ（またはクライ

アント)はPICSカテゴリ値を解釈してプッシュまたはステージングの決定を効率化することができる。

#### 【0027】

【発明の実施の形態】図1に本発明の機能を備えたプロキシ・サーバの階層の全体アーキテクチャを示す。図に示すように、クライアント(600...603)はプロキシ・サーバ(30...55)の階層(レベル0...レベル3)を介してインターネット(25)に接続できる。この例に限って言うと、図示のプロキシ階層は4つのレベルのプロキシ・サーバを含む。プロキシ階層のレベルはいくつでもかまわず、クライアント(600...603)は実際そのどのレベルにも接続できることを当業者なら理解するであろう。最高レベル(レベル0)のプロキシ(30)はインターネットに接続する。レベル1のプロキシは2つ(35と37)あり、一方のレベル1プロキシ35はレベル2のプロキシ(40と43)に接続される。クライアント603はそのすぐ上のレベル(レベル3)のプロキシ(50)に、次いでレベル2、1、0のプロキシ(40、35、30)に結合される。クライアントは、インターネット(25)を介して様々なウェブ・サイト(20)にアクセスすることができる。クライアントはステージングおよびユーザ情報を管理するためそれ自体のクライアント・プロキシを有することができる。プロキシ・サーバの階層は1つのクライアント(600)と単一のプロキシ・サーバを含むこともできることを当業者なら理解するであろう。

【0028】クライアント(603)から見ると、ある種のプロキシ(55、43、および37)はそのプロキシ階層の一部ではない。代表的な通信パスがすぐ上位レベルのプロキシ経由であるが下位レベルのプロキシは直接インターネットの他の上位レベルのプロキシへ通信が可能であることを当業者なら理解するであろう。

【0029】従来のプロキシ階層では、プッシュされたオブジェクトの受信後、オブジェクトは次の(下位の)レベルのプロキシへ下方にプッシュされる。他方、ローカルに使用できないオブジェクトでの「プル」要求の場合、欠落オブジェクトの要求は次の上位レベルのプッシュに対して行われる。上位レベル・プロキシが以前にオブジェクトをステージングした場合、このプロキシはオブジェクトを下方に渡す。そうでない場合、このプロキシは上位レベル・プロキシからオブジェクトを入手しようとする。入力されたオブジェクトはそのオブジェクトを要求している次の下位レベルのプロキシへ下方に渡される。

【0030】図1はオブジェクトおよび要求の情報のフローを表す論理接続だけを示す図であることに注意されたい。この図は物理的接続図ではない。論理接続は作業負荷やノードまたはリンク障害などの物理的イベントによって変わる場合がある。オブジェクトのタイプが異なれば、ユーザへ到達する論理パスも異なる場合がある。

【0031】概観すると、本発明のサーバ(30...50)は、オブジェクトと共にプッシュ・オブジェクトおよびそのステージングの決定に関する情報を次の(下位の)レベルのプロキシへプッシュする。実際、それ自体のステージングの決定だけでなく、上位階層全体のステージングの決定(1つまたは複数の)を下方に渡すことができる。

【0032】本発明は階層の上下方に情報を効率的に送信するための機能をさらに備える。httpを実施する場合、情報交換は既存のウェブ・プロトコルを使ってオブジェクト・ヘッダ内に含めることができる。PICS(「インターネット・コンテンツ選択用プラットフォーム」)は電子コンテンツに関するメタ情報を送信する方法を規定する。PICSはウェブ・コンソーシアム・プロトコル勧告である(<http://www.w3.org/PICS>を参照)。PICSは「このコンテンツにどれくらいの量のnudity(ヌーディティ)が関連付けられているか」などの値ベースの評価ラベルを昇進するために初めて使用されたが、メタ情報のフォーマットと意味は完全に一般的である。PICSでは、電子コンテンツのメタ情報が「評価サービス」または情報の生成源および予定利用状況に従ってグループ分けされそのグループの1つの中で情報の任意の数のカテゴリまたはディメンションを送信できる。各カテゴリには許される値の範囲があり、特定の1件のコンテンツについて、特定のカテゴリが単一または複数の値をとることができる。さらに、メタ情報グループ(「PICSラベル」として知られている)は満了情報を含むことができる。また、PICSが複数の電子コンテンツに適用する機能もある。特定の1件の電子コンテンツのPICSラベルは独立してコンテンツに追加またはそこから削除できる。

【0033】例えば、「評価サービス」フィールドが「Safesurf」評価システムに従って値ベースの評価ラベルを含むことを示す単一のPICSラベルを付けたサーバイメージ・ファイルを送信することができる。本発明によれば、イメージ・ファイルは企業プロキシ通過する際に「評価サービス」フィールドがプロキシ・ステージング情報を含むことを示す第2のPICSラベルを受信することができる。部門プロキシを通過する際に第2のPICSラベルをはがすことができる。このように、クライアント・コンピュータは第1のPICSラベルだけを参照することができる。httpプロトコルはPICSをサポートする要求ヘッダおよび応答ヘッダを増強している。NNTPなどの他の共通アプリケーション・プロトコルを規定する技術機関もPICSサポートを追加することを考慮している。こうしたプロトコルの一部として、要求によって、所望のPICSラベルのタイプのリストを含めることもできる。またPICSは中央のラベル局サーバからPICS情報を受信する照会フォーマットを規定している。PICSラベルの例



は、(PICS-1.1 "http://the.rating.service" label for "http://the.content.exp" 1997.07.01T08:15-0500"r (n4s3v210)) で、ここで「n」、「s」、「v」、「l」は各種メタ情報タイプの送信名、このコンテンツの適用可能な値は4 (nの場合)、3 (sの場合)、2 (vの場合)、および0 (lの場合) である。ID「http://the.rating.service」を認識するソフトウェアだけがこれらのカテゴリおよび値の解釈が可能であろう。

【0034】好ましい実施形態では、3つの異なるPICSラベルが使用される。プッシュ・ラベルまたは(P-ラベル)と呼ばれる第1の種類のPICSラベルはコンテンツ・プロバイダまたは上位レベルのプロキシが使用してオブジェクトの注釈付け、すなわち、プッシュ・オブジェクトの特性を指定したりそれに関する情報を指定する。このラベルは次のカテゴリのいずれの組み合わせも含むことができるがそれだけには限定されない。

【0035】・緊急度カテゴリ:「緊急度」カテゴリの値はオブジェクトを下方にプッシュする緊急度を示す。そのカテゴリ値として「UV」を定義する。

【0036】・サマリ・カテゴリ:「サマリ」カテゴリの値はプッシュ・オブジェクト/ドキュメントの短いサマリを示す。より一般的なケースでは、1つのオブジェクトを複数の詳細レベルで指定できる。このコンテンツ階層は上記の3つ以上のレベル(フル・コンテンツおよびサマリ)からなることができる。例えば、別のタイトル・レベルを含むことができる。ニュース・オブジェクトの場合、「テロリストがショッピング・センタに爆弾をしかけて2名が死亡した。」のサマリとニュースの全コンテンツである「爆弾の爆発」などのタイトルを含むことができる。それぞれの追加のコンテンツ・レベルの別のカテゴリを使ってタイトルなど追加のコンテンツ・レベルをP-ラベルで指定できる。この例には、オブジェクト・タイトルの「タイトル」カテゴリがある。さらに、サマリ・カテゴリに全オブジェクト(すなわち、全コンテンツ)からの異なる緊急カテゴリ値を指定もできる。例えば、より高い緊急度カテゴリ値をサマリに指定できる。

【0037】・グループ・カテゴリ:「グループ」カテゴリの値はオブジェクトの分類を示す。例えば社内報の同報通信の場合、代表的なグループ・カテゴリ値は「企業」、「HR」、「部門」、「競合他社」などを含むことができる。グループ・カテゴリを導入する第1の目的は、ユーザ情報(次節で説明する)をグループ・カテゴリが収集できるようにし、グループ・カテゴリによるプッシュを可能にすることである。

【0038】・チャンネル・カテゴリ:「チャンネル」カテゴリの値は同報通信チャンネルまたはコンテンツ送信源を

示す。例えば、内部企業チャンネル、またはPOINTCAST (http://www.pointcast.com) の商標でPointcast, Inc. から利用できる外部チャンネルである。異なるチャンネルは異なるグループ・カテゴリを備えることができる。

【0039】好ましい実施形態では、1つのグループ・カテゴリおよび1つのチャンネル・カテゴリが形態を簡素化すると議論される。当業者は複数のグループまたはチャンネル・カテゴリあるいはその両方への一般化は容易である一チャンネルごとに別々にグループ単位で統計情報が保持され、プッシュ・フィルタリングの決定がオブジェクト・グループに対して利用状況に基づいてなされるため一ことを理解するであろう。

【0040】ユーザ・ラベル(U-ラベル)と呼ばれる第2の種類のPICSラベルを使って階層の下位のレベルから上方にプッシュ・オブジェクトの利用またはプリファレンス情報を送信できる。このラベルは次のカテゴリのいずれの組み合わせも含むことができるがそれだけには限定されない。

【0041】・利用状況(usage)カテゴリ:「利用状況」カテゴリの値はオブジェクト(オブジェクト・グループ内の)はどのくらいの頻度で下位階層で参照/アクセスされているかを示す。このカテゴリ値は「RV」で示される。

【0042】・プリファレンス・カテゴリ:「プリファレンス」カテゴリの値はユーザがプロファイル指定によって指定する興味がある対象を示す。プロファイル情報はユーザの興味が変わって更新しなくなると廃止できる。このカテゴリ値は「PV」で示される。

【0043】「ステージング」ラベル(C-ラベル)と呼ばれる第3のタイプのPICSラベルはプロキシがオブジェクトが階層を通過する際に動作情報(キャッシング/ステージング情報などの)を送信し、共用するために使用する。このラベルは次のカテゴリのいずれの組み合わせも含むことができるがそれだけには限定されない。

【0044】・ステータス・カテゴリ:「ステータス」カテゴリの値はオブジェクトが上位カテゴリ内でステージングされるかまたはステージングの方法あるいはその両方を示す。このラベルはそれぞれの上位レベル階層でオブジェクトがステージングされるかどうか指定できる。コンテンツ階層が使用できる場合、カテゴリ値は全ドキュメントとサマリのどちらをステージングするかをさらに示すことができる。

【0045】オブジェクトのステージング・ステータスは「CV」で示される。任意のプロキシでCV値を決定する方法の1つを以下に説明する。例えば、2進値表記を使って次のようにCV値を決定できる。n番目のレベルのプロキシの場合、下方に渡されるオブジェクトのCV値はn個のビットを備え、k番目のビットには(n-

k) レベルのプロキシが階層内を下方にオブジェクトを送信した際にオブジェクトをステージングした場合は 1 となる。そうでない場合、k 番目のビットは 0 になる。さらに、ステージング・ステータス情報を使って、オブジェクトをすぐ上位のレベルのプロキシに要求せずに、オブジェクト要求をオブジェクトをステージングした可能性が高い最も近い上位のレベルのプロキシへ転送することもできる。

【0046】図 1 を再度参照して、例えばオブジェクト A の CV 値を考える。レベル 0 プロキシ (30) とレベル 2 プロキシ (40) はオブジェクト A をステージングし、レベル 1 プロキシ (35) はオブジェクト A をステージングしていないものとする。レベル 3 プロキシ 50 へ下方に渡されたオブジェクト A の CV 値はの場合「101」(2 進値) または 5 (10 進値) になる。プロキシ 35 オブジェクトプロキシ 40 へ下方に渡されたオブジェクト A の CV 値はそれぞれ「1」と「10」である。階層内でステージング・ステータスを表す別の方法もあることを当業者は理解するであろう。これより簡素ではあるが正確さが低い方法として、単一のビットを使っていずれかの上位レベルのプロキシがオブジェクトをステージングしたかどうかを表す方法がある。ビットがオンの場合、上位レベル・プロキシがオブジェクトをステージングしている。そうでない場合、どの上位レベルのプロキシもオブジェクトをステージングしていない。

【0047】図 2 に図 1 のプロキシ・サーバのアーキテクチャの詳細例を示す。慣例として、プロキシ・サーバは CPU (200)、ディスク (205)、持続性データまたはプログラム/コード記憶用の磁気、電子、または光記憶メディア、および CPU (200) によるデータまたはプログラムあるいはその両方の動的アクセスまたは実行あるいはその両方を行うためのメモリ (207) を含む。本発明の精神と範囲を逸脱することなく、メモリ (207) 内にインスタンス化された 1 つまたは複数のコンポーネントがディスク (205)、ネットワーク (25) 他のサーバから直接アクセスして保守でき、複数のサーバに配布できることを当業者は理解するであろう。好ましくは CPU (200) 上で実行可能なソフトウェアとして具体化されたプロキシ・サーバの 3 つの主要コンポーネントは、ブッシュ・ハンドラ (220)、欠落ブッシュ・オブジェクト要求ハンドラ (240)、およびヘッド要求ハンドラ (260) である。これらのコンポーネントについてはそれぞれ図 5、10 および 4 を参照しながら以下に詳述する。

【0048】メモリ (207) は本発明の機能に関連したいくつかのその他の構造を含む。図 8 を参照しながら以下に詳述するように、キャッシュ (270) は各プロキシ・ノードで保守される。ブッシュがフィルタリングされた場合にキャッシュを使ってブッシュ・オブジェクトをステージングする。オブジェクトを後に要求する場合、この手順でアクセス時間が短縮される。キャッシュがディスク (205) などの記憶階層の下位レベルまで拡張できることに注意されたい。したがって、プロキシ内のキャッシングまたはステージングされたオブジェクトは階層のどのレベルにも常駐できる。図 4 を参照しながら以下に詳述するように、それぞれの次のレベルのノードの総合ユーザ情報はユーザ情報 280 が示すように保守されてフィルタリングの決定を支援する。図 8 を参照しながら以下に詳述するように、それぞれのステージングされたオブジェクトに関連する一定のオブジェクト情報 290 (UV などの) はフィルタリングの決定のために保守される。このことは、図 10 を参照しながら以下に詳述するように、ステージングされたオブジェクトをバージするかどうかについての後の決定に影響する。

【0049】図 3 に本発明の機能を備えたプロキシ・サーバ論理の例を示す。図示のように、ステップ 310 でプロキシ・サーバは入力待ち状態となる。ステップ 315 で、受信した入力によって、処置が分岐する。受信した入力 (次の上位レベルから) がブッシュの場合、ステップ 320 でブッシュ・ハンドラ 320 が呼び出される。ブッシュ・ハンドラの詳細例は図 5 を参照しながら後述する。ステップ 330 で、受信した入力欠落ブッシュ・オブジェクト要求の場合、欠落ブッシュ・オブジェクト要求ハンドラ 240 がステップ 335 で呼び出される。欠落ブッシュ・オブジェクト要求ハンドラ 240 は上位レベル・プロキシ・ノードによってフィルタリングされたオブジェクトに関して下位レベル・プロキシ (またはクライアント) からの要求を処理する。欠落ブッシュ・オブジェクト要求ハンドラの詳細例は図 10 を参照しながら後述する。ステップ 350 で、受信した入力欠落要求の場合 (下位レベル・プロキシまたはクライアントからの)、ステップ 360 でヘッド要求ハンドラ 260 が呼び出される。ヘッド要求ハンドラ 260 は HTTP ヘッド要求を処理して下位レベル・プロキシからのユーザ情報を送信する。ヘッド要求ハンドラの詳細例は図 4 を参照しながら後述する。ステップ 350 で、本発明の対象ではない他のタイプの入力について (従来の HTTP 「プル」 要求または FTP 要求など)、該当するミセレニアス・ハンドラ (370) が呼び出される。

【0050】図 4 にヘッド要求ハンドラ (260) の例を示す。図示のように、ステップ 410 でプロキシは次の下位レベル・ノード i から受信したヘッド要求がヘッダ内にユーザ・ラベル (U-ラベル) を含むかどうかチェックする。U-ラベルは関連するカテゴリ値がそれぞれ RV と PV で表される利用状況およびプリファレンスの 2 つのカテゴリを含む。次の下位レベル・ノード i のそれぞれについて、プロキシ・サーバはメモリ内にその利用状況とプリファレンス・カテゴリ値をユーザ情報 (280) の RV (i) と PV (i) 内にそれぞれ保守

する。ステップ420で、RV(i)およびPV(i)はノードiの新たに受信した値に合わせて更新される。ステップ430で、プロキシ・ノードは総合利用状況とプリファレンス値(それぞれRVnodeおよびPVnodeで示されている)をすべての次の下位レベルのノードにわたって保守する。これらの2つの平均方法を更新する際は指数平均法を用いることが好ましい。より詳細に言えば、PVnodeはその現在の値プラス新しいPV(i)と旧PV(i)値の差の小数部分に設定される。この一例は小数部分が0.5である。RVnode値も同様に設定される。ステップ440で、HTTPヘッダ中の他の情報を処理することができる。この一例はオブジェクトが最後に変更された時点に基づくオブジェクトの「新鮮度」をチェックすることである。

【0051】図5にプッシュ・ハンドラ220の例を示す。図示のように、ステップ510で次の(上位)レベルからプッシュされたオブジェクトが全オブジェクトの場合、すなわち、ヘッダだけではなく全コンテンツの場合、ステップ515でプッシュ・オブジェクト・フィルタリング・ルーチンが呼び出され、次の下位レベル・ノード(1つまたは複数)へのプッシュのフィルタリング決定がなされる。プッシュ・オブジェクト・フィルタリング・ルーチンの詳細例は図6を参照しながら後述する。ステップ520で、このノードのオブジェクトをステージングするかどうか決定するため、ステージング決定ルーチンが呼び出される。ステージング決定ルーチンの詳しい例については、図8に関して述べる。ステップ510で、上位レベル・プロキシからサマリ情報だけがプッシュされた場合、ステップ525でプッシュ・サマリ・フィルタリング・ルーチンが呼び出され、サマリを下位レベル・ノードへプッシュするかどうか決定される。プッシュ・サマリ・フィルタリング・ルーチンの詳細例は図7を参照しながら後述する。ステップ530でプッシュ実行ルーチンが呼び出されて下位レベル・ノードへのプッシュが実行される。プッシュ実行ルーチンの詳細例は図11を参照しながら後述する。

【0052】図6にプッシュ・オブジェクト・フィルタリング・ルーチンの例を示す。ステップ610で、インデックス変数iが0に初期化される。ステップ610で、インデックス変数iが次の下位レベルのノードの数Nより小さい場合、ステップ615でiの値が例えば1インクリメントし、プッシュ決定変数P(i)が例えば0に初期化される。ステップ620で、オブジェクトのプッシュ(UV)の緊急度レベルが緊急プッシュしきい値(PTH)より大きく、PV(i)が0より大きい場合(複数の下位レベルユーザがオブジェクト内にプロファイルなどによってインタレストを指定した場合)、ステップ660でプッシュ決定が選択され(P(i)が1に設定され)、全オブジェクトがノードiへプッシュされる(図11のステップ1120を参照)。そうでない

場合、ステップ630で、(論理)関数F(RV(i), PV(i), UV)、すなわち、オブジェクトのプロパティ(例:UV)と、オブジェクトユーザ特性(例:RV(i)およびPV(i))の関数が呼び出され、フィルタリング決定を行う。F(RV(i), PV(i), UV)が真の場合、ステップ660が実行されてP(i)は1に設定され、全オブジェクトがノードiへプッシュされる。F(RV(i), PV(i), UV)の簡素な例は論理式((RV(i)UV>QTH) and (PV(i)>0))である。ただし、QTHは1などのしきい値である。より複雑なF関数を設計してオブジェクトの帯域幅またはサイズあるいはその両方の因数を考慮できる。例えば、別のF(RV(i), PV(i), UV)を((RV(i)UV>WTH) and (PV(i)>0))の論理式で表すことができる。ただし、WTHはオブジェクト・サイズと共に増加し、利用可能な帯域幅と共に減少するしきい値である。WTHはドキュメントの満了時間を考慮することもできる。この値を満了時間が長いオブジェクトに関して低く設定できる。ステップ640で、(論理)関数G(RV(i), PV(i))が呼び出され、プッシュ・サマリ決定が実行される。G(RV(i), PV(i))の例は論理式((RV(i)>0) and (PV(i)>0))である。F関数と同様に、より複雑なG関数を設計して帯域幅などの他の因数をプッシュ・サーバの決定で考慮することができる。ステップ650で、P(i)は0.5に設定され、サマリは下方にノードiへプッシュされる。

【0053】図7にプッシュ・サマリ・フィルタリング・ルーチンの例を示す。ステップ710で、インデックス変数iは0に初期化される。ステップ720で、インデックス変数iが次の下位レベルのノードの数Nより小さい場合、ステップ730でiの値が例えば1インクリメントする。ステップ740で、関数G(RV(i), PV(i))が呼び出され、プッシュ・サマリ決定が実行される(これはステップ640に呼び出される関数と同じ関数である)。ステップ740で、G関数が真の場合、ステップ750でP(i)は0.5に設定され、サマリは下方にノードiへプッシュされる(図11のステップ1140を参照)。

【0054】図8にステージング決定ルーチンの例を示す。ステップ810で、オブジェクトOの「ステージング緊急度」因数が計算される。この因数はCacheU(O)として表される。オブジェクトのステージング緊急度の計算論理の例は図9を参照しながら以下に詳述する。ステップ815で、CacheU(O)の値は上位レベル・プロキシにあるオブジェクトOのステージング・ステータスに基づいて下方に調整できる。ステージング・ステータス情報はHTTPヘッダのステージング・ラベル(C-ラベル)に指定される。オブジェクトOがどこかの上位レベル・プロキシ・キャッシュですでにス

ステージングされている場合、それを現在のノードでステージングする必要は減少する。ステップ 8 2 0 で、Cache U (O) が 0 より大きい場合、ステップ 8 3 0 で、空間量 (S) が (1) オブジェクト O よりもステージング緊急度が低いすべてのオブジェクト O<sub>j</sub> によって占められているか、(2) 利用可能、すなわち、現在使用されていないかが判定される。ステップ 8 4 0 で S がオブジェクト O のサイズ (O) より大きいと判定された場合、ステップ 8 5 0 でオブジェクト O の CV 値が更新されて現在のノードでステージングされたことを示す。詳細に言えば、元の CV 値は 2 倍して 1 を加えて新しい CV 値とすることができる。ステップ 8 7 0 で、オブジェクト O はキャッシュ (2 7 0) に記憶され、オブジェクトに関連する CV、UV、および Cache U の値はメモリのオブジェクト情報 (2 9 0) 部分に記憶される。オブジェクト O は必要に応じて、ステージング緊急度がより低い値を持つ他のオブジェクトと交換できる。ステップ 8 2 0 で Cache U (O) の値がゼロの場合、オブジェクトはステージングされず、CV 値は更新されてこのプロキシでステージングされていないことを示す。具体的には、元の CV 値に 2 を掛けて新しい CV 値を得ることができる。さらに、オブジェクトの C-ラベルはステップ 8 5 0 またはステップ 8 8 0 で生成された新しい CV 値をとり、オブジェクトが階層中を下方にプッシュされる際のステージング・ステータスを示す (図 11 のステップ 1 1 0 0 を参照)。

【0 0 5 5】本発明の精神と範囲を逸脱することなく、さまざまなキャッシュ組込み処理が可能であることを当業者は理解するであろう。例えば、今まで要求がなかったとしても新しいカテゴリ内のオブジェクトを空間変数単位でステージングすることができる。

【0 0 5 6】図 9 にオブジェクト O のステージング緊急度の計算論理の例を示す。ステップ 9 0 5 で、ステージング緊急度の値 Cache U (O) が例えば 0 に初期化される。ステップ 9 1 0 で、インデックス変数 i が 0 に初期化される。ステップ 9 2 0 で、インデックス変数 i が次の下位レベルのノードの数 N より小さく、ステップ 9 3 0 でプッシュ決定変数 P (i) が 1 でない場合、ステップ 9 4 0 で Cache U (O) は RV (i) UV (O) だけインクリメントする。ステップ 9 4 5 で、i はインクリメントし、処理はステップ 9 2 0 に戻る。ステップ 9 2 0 で i > N の場合、処理は終了する。

【0 0 5 7】図 10 に欠落プッシュ・オブジェクト要求ハンドラ (2 4 0) の例を示す。プッシュ・オブジェクト O がフィルタリングされ、その後下位レベル・ノードから要求された場合、ステップ 1 0 0 5 でオブジェクト O が現在のノードでステージングされているか判定される。ステージングされている場合、ステップ 1 0 1 0 でオブジェクトはその C-ラベルに CV 値を挿入して要求されたノードへ返送される。ステップ 1 0 2 0 で、オブ

ジェクト O のステージング緊急度の値が再び計算される。ステップ 1 0 3 0 で、この値がゼロまで低下した (オブジェクト上のインタレストを持つすべての次の下位レベルのノードがそのオブジェクトのコピーを受信した) 場合、または何か別の規定値もしくは計算値にまで低下した場合、ステップ 1 0 4 0 でオブジェクト O はもはやステージングされない。ステップ 1 0 0 5 で、オブジェクトがステージングされない場合、ステップ 1 0 8 0 で要求は上位レベル・プロキシ・サーバまたはコンテンツ送信元へ転送される。

【0 0 5 8】図 11 にプッシュ実行ルーチンの例を示す (図 5、ステップ 5 3 0)。ステップ 1 1 0 0 で、オブジェクトの CV 値 (図 8 のステップ 8 5 0 または 8 8 0 から) が HTTP ヘッダの C-ラベルに挿入される。ステップ 1 1 2 0 で、オブジェクト O 全体が P (i) が 1 に等しいすべての次の下位レベルのノードへプッシュされる。ステップ 1 1 4 0 で、オブジェクト O のサマリ・ヘッダが P (i) が 0.5 のすぐ下のレベルのすべてのノードにプッシュされる。ステップ 1 1 6 0 で、あるノード i へのプッシュが不成功の場合 (リンクまたはノード障害もしくはモバイル・クライアントがアクセス不能)、P (i) 値がゼロにリセットされる。さらに、オブジェクト O のステージング決定ルーチン (図 5 のステップ 5 2 0) がチェックされる。ステージング決定出力がオブジェクトのステージングを実行しない場合、ステージング決定ルーチン (図 8) は新しい P (i) 値のセットを指定して再呼び出され、プッシュの一部が失敗したためにオブジェクトを今ステージングするかどうかを決定する。

【0 0 5 9】プロキシの一部がフィルタリング・プロトコルに適合せずコラボレーションに参加しない従来のプロキシである異機種プロキシ・サーバ環境では動的プッシュ・フィルタリングが有効であることを当業者は理解するであろう。

【0 0 6 0】本発明の好ましい実施形態で、ウェブ・サーバの一般のプッシュ・フィルタリング方式について説明してきた。ただし、本発明はプッシュ対象のオブジェクトが同様の特性を備えるいかなる種類の状況にも適用可能で、必ずしもインターネットや WWW の適用分野に限定されないことを当業者は理解するであろう。

【0 0 6 1】本発明の好ましい実施形態は階層の親ノードおよび子ノード間のステージングを伴う協動的なプッシュ・フィルタリングであるが、兄弟ノード間のコラボレーションを含めるように容易に適合させることができる。例えば、上位階層でプロキシが要求されたオブジェクトをステージングしていない場合、プロキシから兄弟プロキシへ照会ができる。さらに、図 8 のステージング決定を、兄弟ノードを含め、ただしこれに限定されず、階層外のプロキシ・ノードのステージング決定の因数に容易に適応させてそのステージング決定に使用すること

ができる。

【0062】まとめとして、本発明の構成に関して以下の事項を開示する。

【0063】(1) プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてその後プッシュされたオブジェクトを動的にフィルタリングする方法であって、プッシュされたオブジェクトに関連する利用状況情報を階層中を上方に送信するステップと、送信されたオブジェクト利用状況情報に基づいてその後プッシュされたオブジェクトをフィルタリングするステップとを含む方法。

(2) プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてプッシュされたオブジェクトを動的にフィルタリングする方法であって、情報を総合しノード間で交換するステップと、プッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングするステップとを含む方法。

(3) 前記フィルタリング・ステップがフィルタリングされたオブジェクトに関連するメタ情報を送信するステップをさらに含む、上記(2)に記載の方法。

(4) フィルタリング決定、選択された下位レベル・プロキシ/クライアントへのプッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップをさらに含む、上記

(1)に記載の方法。

(5) プッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後プッシュされたオブジェクトをフィルタリングする前記ステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む、上記

(1)に記載の方法。

(6) メタ情報を送信する前記ステップが、オブジェクトの短い説明を階層中を下方に送信するステップと、フィルタリングされたオブジェクトをプロキシ階層内でステージングするステップとをさらに含む、上記(5)に記載の方法。

(7) オブジェクトをステージングする前記ステップが、ユーザ・プロファイルで示された潜在的なインタレストを有するすぐ下位のレベルのすべてのプロキシまたはクライアント・ノードがオブジェクトを受信したか、あるいはステージング緊急度因数が所定および計算しきい値を下回ったか、そのいずれかの後にステージングされたオブジェクトをバージするステップをさらに含む、上記(4)に記載の方法。

(8) 前記フィルタリング・ステップがさらに階層内のすべての下位クライアント・ノードによるプッシュされたオブジェクトの総合利用状況およびプリファレンスを規定するステップをさらに含む、上記(1)に記載の方法。

(9) オブジェクトがオブジェクト・グループに分類さ

れ、各オブジェクト・グループの利用状況情報が以前のユーザ要求パターンに基づく、上記(4)に記載の方法。

(10) 利用状況情報が下位レベル・プロキシまたはクライアントの要求パターンを含み、前記プッシュされたオブジェクトを異なるグループに分類するステップと、プッシュされたオブジェクトのグループ分けを含む利用状況情報を総合しノード間で交換するステップと、総合され交換された情報およびプッシュされたオブジェクトのグループ化に基づいてプッシュされたオブジェクトをフィルタリングするステップとをさらに含む、上記

(1)に記載の方法。

(11) フィルタリング決定またはステージング決定が帯域幅、オブジェクト・プロパティ、またはクライアント特性のいずれかの関数である、上記(4)に記載の方法。

(12) クライアント特性がユーザ・プロファイルまたはプリファレンス情報を含む、上記(11)に記載の方法。

(13) オブジェクト緊急度標識をプッシュされたオブジェクトに関連付けて階層中を下方に送信するステップをさらに含む、前記フィルタリング・ステップがオブジェクト緊急度標識の関数である、上記(1)に記載の方法。

(14) 緊急度標識を1つまたは複数の異なる階層レベルに関連付けるステップをさらに含む、前記ステージング・ステップが、緊急度標識の関数としてコンテンツ階層の1つまたは複数のレベルでオブジェクトをステージングするステップを含む、上記(4)に記載の方法。

(15) 前記フィルタリング・ステップとステージング・ステップのいずれかがオブジェクト・サイズの関数である、上記(4)に記載の方法。

(16) 前記フィルタリング・ステップと前記ステージング・ステップのいずれかがオブジェクトの寿命または満了時間の関数である、上記(4)に記載の方法。

(17) 前記ステージング・ステップにตอบสนองして、プッシュされたオブジェクトのステージング・ステータスをプッシュされたオブジェクトへ送信するステップをさらに含む、上記(4)に記載の方法。

(18) プロキシ階層が、前記フィルタリング・ステップと前記ステージング・ステップのいずれかが階層内のすべてのサーバによっては実行されない異種プロキシ階層を含む、上記(4)に記載の方法。

(19) PICS利用状況ラベルを作成し、PICSカテゴリ値を使ってプロキシ階層の下位レベルでの総合利用状況情報を表すステップをさらに含む、前記送信ステップがPICS利用状況ラベルを使って総合利用状況情報を階層中を上方に送信するステップを含む、上記

(2)に記載の方法。

(20) PICSステージング・ラベルを作成し、P I

CSカテゴリ値を使ってプロキシ階層の所与のレベルにおけるステージングされたオブジェクトのステージング・ステータスを表すステップと、PICSステージング・ラベルを使ってステージング・ステータスを階層中を下方に送信するステップをさらに含む、上記(4)に記載の方法。

(21) PICSプッシュ・ラベルを作成し、PICSカテゴリ値を使ってプッシュされたオブジェクトの緊急度標識を表すステップと、PICSプッシュ・ラベルを使って緊急度標識を階層中を下方に送信するステップとをさらに含む、上記(4)に記載の方法。

(22) メタ情報プロトコルを使って階層中を情報を送信する、上記(1)ないし(3)、(5)、(6)、

(9)ないし(10)、(14)ないし(15)、(19)ないし(20)または(21)のいずれか一項に記載の方法。

(23) PISCプロトコルを使って階層中を情報を送信する、上記(1)ないし(3)、(5)、(6)、

(9)ないし(10)、(14)ないし(15)、(19)ないし(20)または(21)のいずれか一項に記載の方法。

(24) PICSプッシュ・ラベルを作成し、PICSカテゴリ値を使ってプッシュされたオブジェクトのサマリを表すステップと、PICSプッシュ・ラベルを使ってサマリを階層中を下方に送信するステップとをさらに含む、上記(4)に記載の方法。

(25) 階層外のプロキシ・ノードについてのステージング決定に基づいてオブジェクトの適応ステージングを行うステップをさらに含む、上記(1)に記載の方法。

(26) プッシュされたオブジェクトのストリームが階層中を下方に送信されるプロキシ階層において、その後プッシュされるオブジェクトを動的にフィルタリングする方法であって、プッシュ・ストリームにメタ情報の注釈を付けるステップと、前記の注釈付けステップに回答

して、1つまたは複数のプッシュされたオブジェクトをフィルタリングするステップとを含む方法。

(27) フィルタリング決定とステージング決定のいずれかが、コンテンツ階層のレベルが異なると変わる、上記(6)に記載の方法。

【図面の簡単な説明】

【図1】本発明の機能を備えた上位レベル・クライアント・サーバ・アーキテクチャの図である。

【図2】図1のサーバの例を示す図である。

【図3】サーバ論理の例を示す図である。

【図4】ヘッド要求ハンドラの例を示す図である。

【図5】プッシュ・ハンドラの例を示す図である。

【図6】プッシュ・オブジェクト・フィルタリング・ルーチンの例を示す図である。

【図7】プッシュ・サマリ・フィルタリング・ルーチンの例を示す図である。

【図8】ステージング決定ルーチンの例を示す図である。

【図9】ステージング緊急度の計算例を示す図である。

【図10】欠落したプッシュ・オブジェクト要求ハンドラの例を示す図である。

【図11】プッシュ実行ルーチンの例を示す図である。

【符号の説明】

20 ウェブ・サイト

25 インターネット

30 レベル0 プロキシ

35 レベル1 プロキシ

37 レベル1 プロキシ

40 レベル2 プロキシ

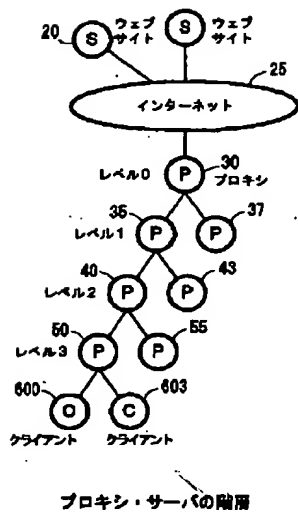
30 43 レベル2 プロキシ

50 レベル3 プロキシ

600 クライアント

603 クライアント

【図 1】



【図 2】

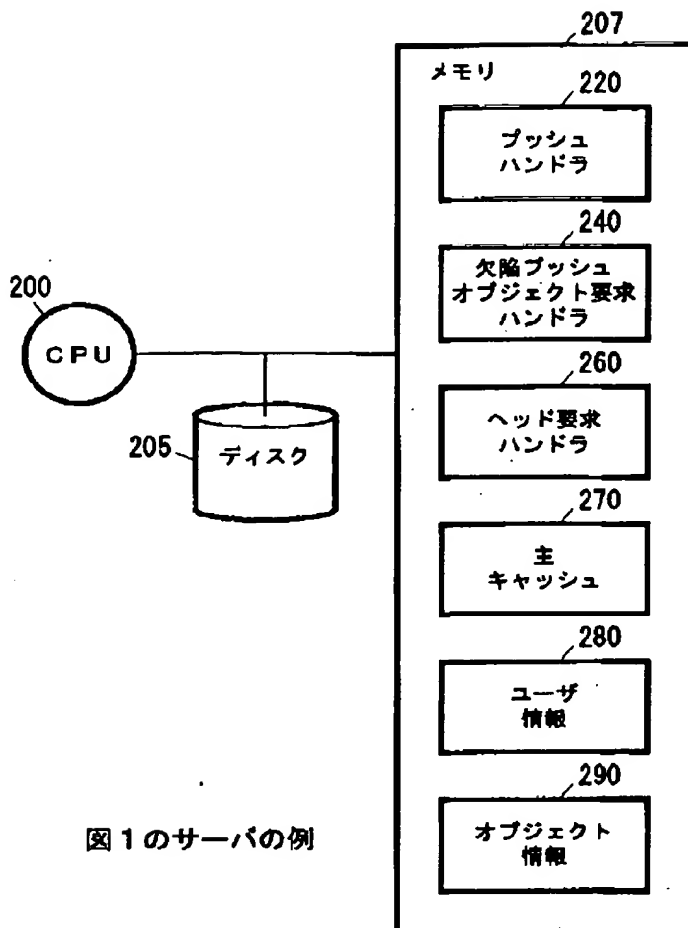
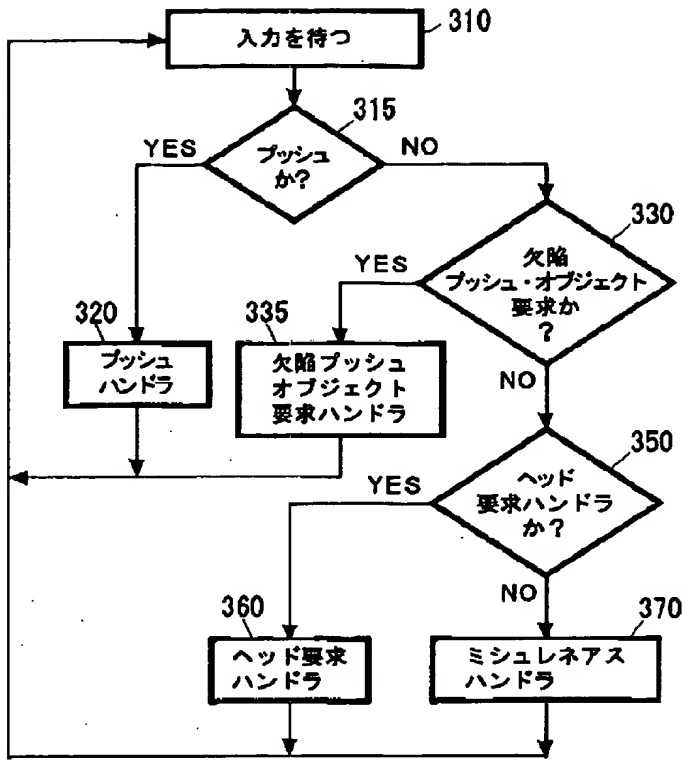


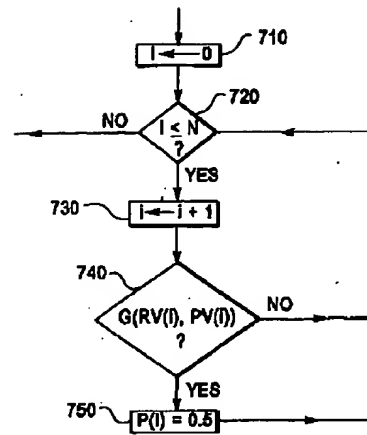
図 1 のサーバの例

【図3】



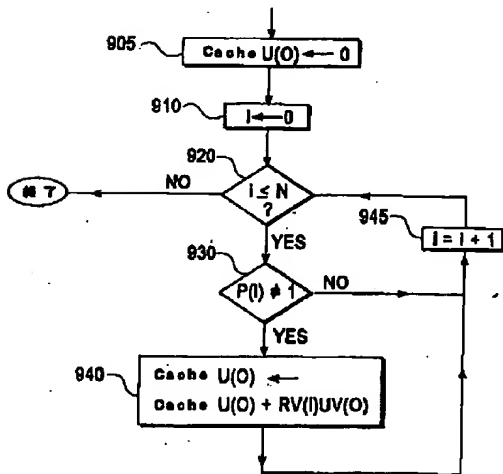
サーバ論理の例

【図7】



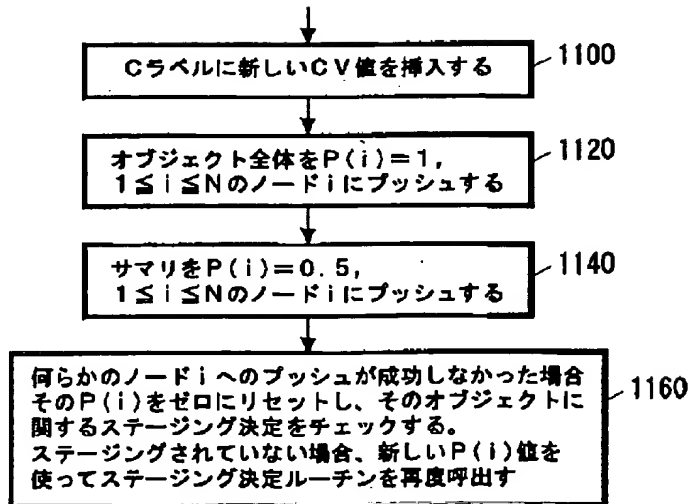
プッシュ・サマリ・フィルタリング・ルーチン

【図9】



ステージング緊急度の計算

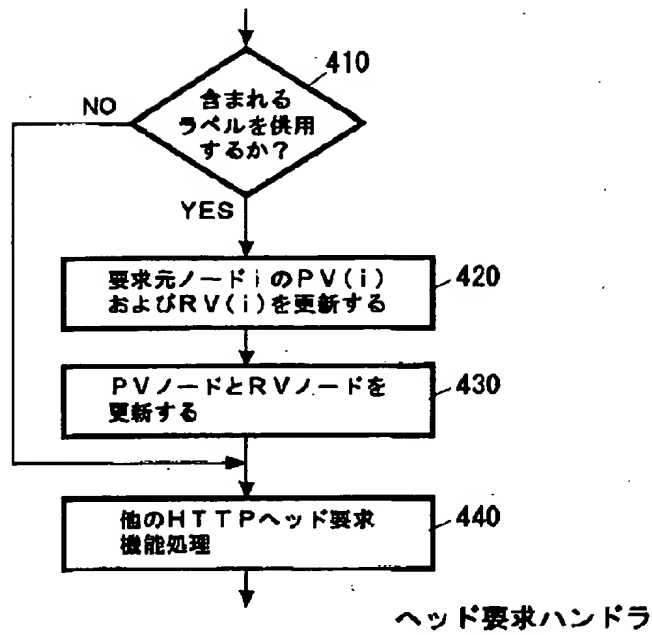
【図11】



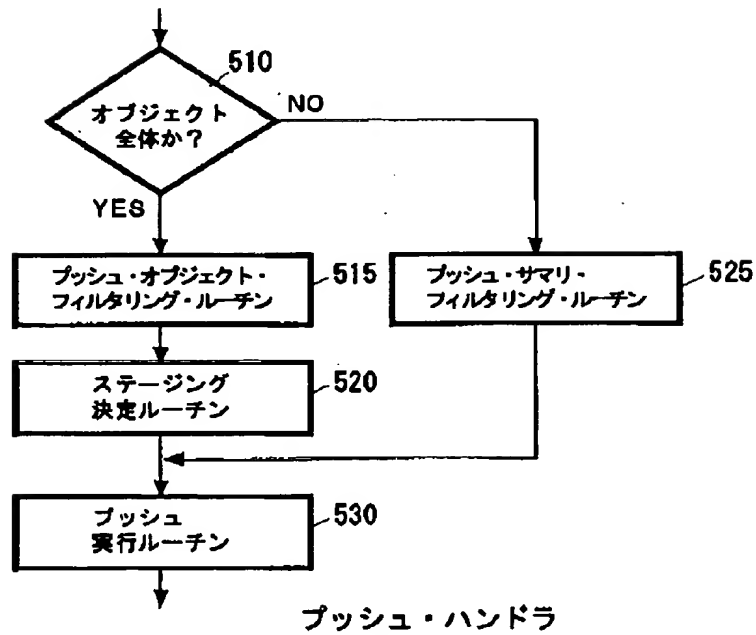
プッシュ実行ルーチン



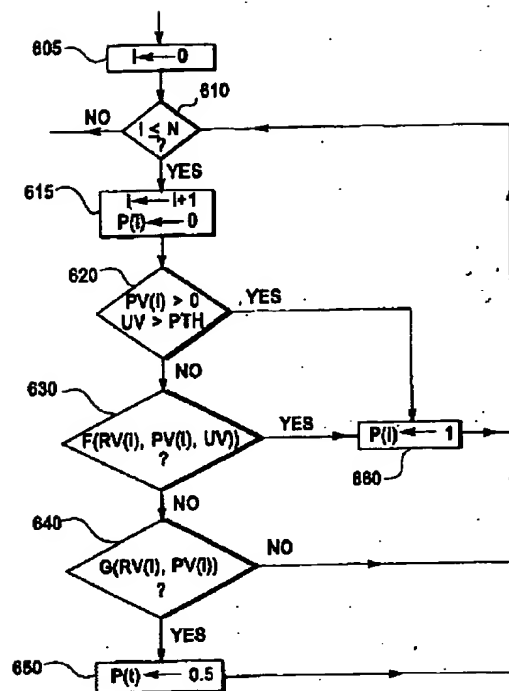
【図 4】



【図 5】

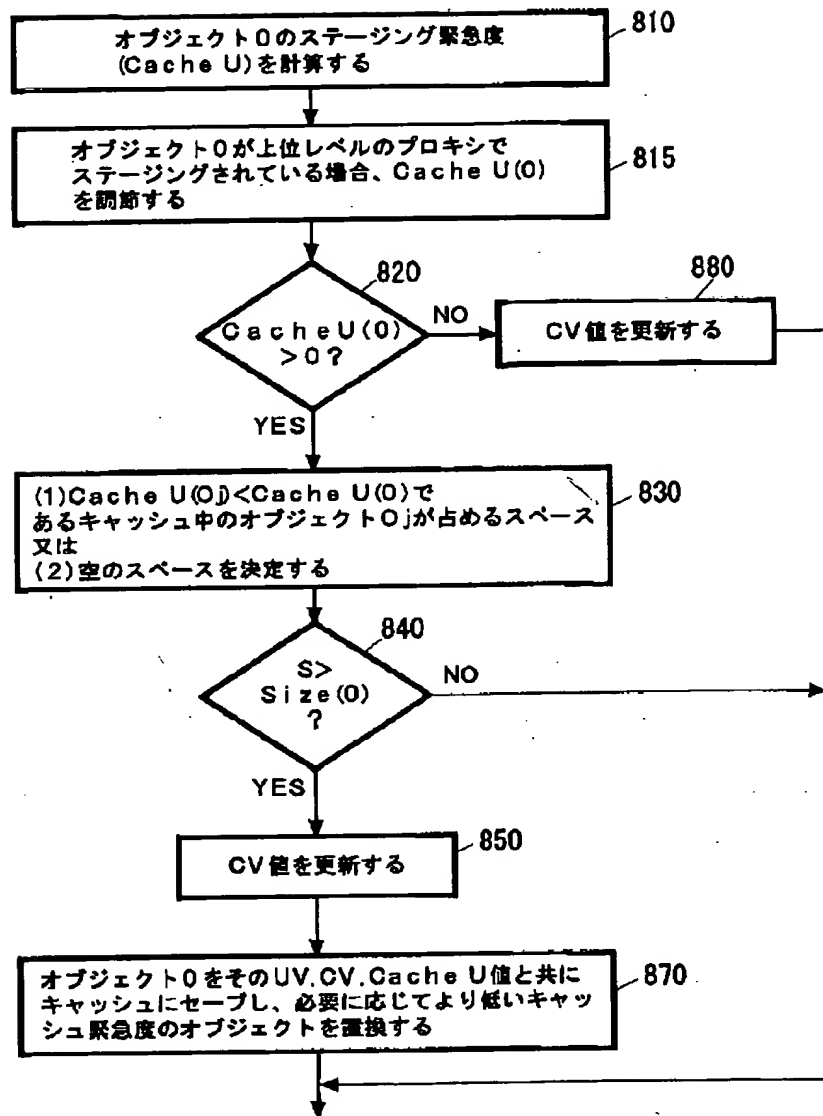


【図 6】



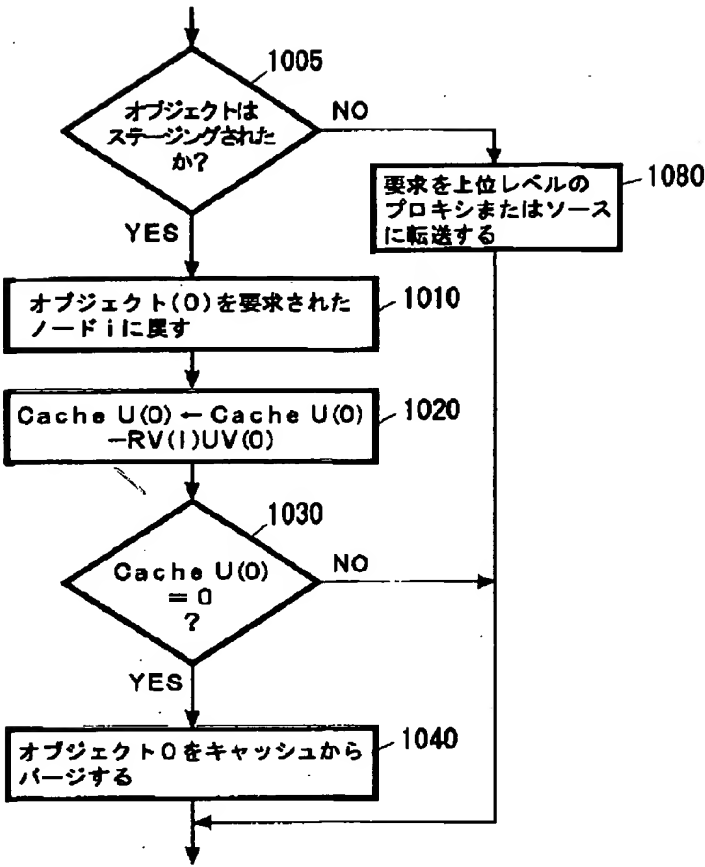
プッシュ・オブジェクト・フィルタリング・ルーチン

【図 8】



ステージング決定ルーチン

【図 1 0】



欠陥 プッシュ・オブジェクト要求ハンドラ

フロントページの続き

(72)発明者   ピーター・ケニス・マルキン  
              アメリカ合衆国10502   ニューヨーク州ア  
              ーズレイ   ブランブル・ブルック・ロード  
              64

(72)発明者   ロバート・ジェフリー・シュロス  
              アメリカ合衆国10510   ニューヨーク州ブ  
              ライアークリフ・マナー   ホルブルックD  
              レーン   155

(72)発明者   フィリップ・シーニルン・ユー  
              アメリカ合衆国10514   ニューヨーク州チ  
              ャバクア   ストーノウエイ   18